La presentación

Pere Barnola Augé

P08/93133/01507



Índice

Introducción

Como ya hemos ido diciendo en los capítulos anteriores, una de las claves para construir un documento web de forma correcta es separar el contenido de como éste se presenta. En el contenido, sólo tendrá que haber las etiquetas que describen el contenido, pues no requiere ninguna etiqueta de representación o de estilos para transmitir completamente su mensaje. El hecho de que la presentación vaya por otra parte nos facilitará modificar la visualización del contenido en función del medio en que lo queramos mostrar. Por lo tanto, nos dará mucha flexibilidad al mismo tiempo de modificar el "look and feel" de una web. Por ejemplo, si el contenido se tiene que ver en un teléfono móvil, es lógico que modifiquemos el diseño para mejorar la legibilidad y usabilidad en este entorno, aunque el contenido sea el mismo.

Contenido complementario

CSS son las siglas de *cascading style sheets*, es decir, hojas de estilo en cascada.

La tecnología que nos permitirá dar estilo, con cierta facilidad, a nuestro documento web serán las **CSS**. En los próximos capítulos las veremos en profundidad.

CSS Zen Garden

En la web siguiente, podemos ver cómo un mismo contenido se puede visualizar de manera completamente diferente sólo con el hecho de modificar el estilo: http://www.csszengarden.com

1. Las hojas de estilo (CSS)

Las hojas de estilo en cascada (*cascading style sheets*) son un mecanismo simple que nos describe cómo se muestra un documento en la pantalla del ordenador, o cómo se imprime, o incluso cómo se pronuncia la información del documento en un dispositivo de lectura.

Con las CSS, tendremos un control total sobre el estilo y el formato de nuestros documentos. Cualquier cambio realizado en el estilo establecido para un elemento afectará a todas las páginas vinculadas a la CSS en las cuales aparezca este elemento.

2. Cómo se puede dar estilo a un documento XHTML

La CSS funciona con declaraciones sobre el estilo de uno o más elementos. Por lo tanto, las hojas de estilo están compuestas por una o más declaraciones referidas a elementos de un XHTML. Una declaración (regla) tiene dos partes: una propiedad y el valor de la propiedad.

Ejemplo

`p {font-size: 10em;}` en este caso `p` es el selector y `{font-size: 10em;}` la declaración.

El selector es el que nos hace de enlace entre el documento y el estilo, especificando qué elementos se verán afectados por la declaración. Y la declaración es la parte que nos dice cómo se ven afectados los elementos. En el ejemplo, estaríamos indicando que todos elementos p se verán afectados por la declaración que establece que la propiedad *font-size* tendrá el valor 10em para todos los elementos *p* del documento o documentos que estén vinculados a la hoja de estilo.

A continuación, veremos las formas en que se puede hacer esta vinculación y todas las propiedades que podemos tener en una declaración.

2.1. La vinculación CSS-XHTML

Hay diferentes maneras de dar estilo a un documento mediante CSS, aunque hay algunas que rompen con la premisa de separar el contenido de la presentación.

La mejor manera de hacerlo, y la más recomendable, es utilizando una hoja de estilo externo que se vinculará al documento mediante el elemento `<link>`, el cual tiene que estar situado dentro de la sección `<head>`. En este caso, la hoja de estilo estará en un archivo con extensión ".css" totalmente separado del "xhtml". Lo cual nos permitirá que, al modificar el archivo .css, se modifiquen los estilos de todos los documentos donde está vinculado.

Lo primero será crear un archivo donde colocaremos nuestros estilos. Este fichero lo iremos construyendo a medida que expliquemos las propiedades principales y más utilizadas de css (el resto de propiedades las dejaremos como referencia). Lo guardaremos" como "estilos.css" en el mismo lugar donde tenemos el "index.html" que creamos en el módulo 1. Una vez creado, abriremos nuestro documento XHTML y añadiremos el siguiente código a la cabecera.

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN"
```

```
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="es" lang="es">
<head>
<meta http-equiv="Content-Type" content="text/html; charset=UTF-8" />
<meta name="author" content="Pere Barnola"/>
<meta name="keywords" content="garrotxa, gastronomía"/>
<title>Introducción a la gastronomía de la Garrotxa</title>

<link href="estilos.css" rel="stylesheet" type="text/css" />
</head>
<body>
```

De esta manera ya tendremos realizada la vinculación entre el archivo "estilos.css", que es donde tendremos todos los estilos de nuestro documento, y el archivo "index.html", que es donde teníamos el contenido.

Como hemos dicho, hay otras formas de dar estilos a un documento, aunque rompen ligera o completamente la premisa de separar el contenido de los estilos.

Se puede utilizar el elemento `<style>` en el interior del documento al cual se quiere dar estilo, y que generalmente se situará dentro de la sección `<head>`. Se opta por hacerlo de esta manera cuando los estilos se utilizan sólo en el documento en concreto.

Aquí tenemos un ejemplo:

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="es" lang="es">
<head>
<meta http-equiv="Content-Type" content="text/html; charset=UTF-8" />
<meta name="author" content="Pere Barnola"/>
<meta name="keywords" content="garrotxa, gastronomía"/>
<title>Introducción a la gastronomía de la Garrotxa</title>
<Style type="text/css">
body {
padding-left: 11em;
font-family: Georgia, "Times New Roman", Serif;
color: red;
background-color: #d8da3d;
}
h1 {
font-family: Helvetica, Geneva, Arial, Sans-serif;
}
</Style>
```

```
</head>
<body>
```

También se podrían utilizar los estilos directamente sobre aquellos elementos que lo permiten por medio del atributo `<style>` dins del `<body>`. Sin embargo, este tipo de definición sí que rompe totalmente con las ventajas que nos ofrece el hecho de no mezclar el contenido con la presentación.

2.2. Selectores y declaraciones

Como hemos avanzado en la primera parte, los CSS constan de reglas o declaraciones que relacionamos con los elementos de nuestro documento XHTML mediante los selectores.

Ejemplo

En este ejemplo, `p{color: red;}`, el selector `p`, le dice al navegador la parte del documento que se verá afectado por la regla.

Los selectores pueden aparecer individualmente o agrupados, separándolos con comas:

Por tanto, esto

```
p, h1, h2 {
color: red;
}
```

sería lo mismo que:

```
p {color: red;}
h1 {color: red;}
h2 {color: red;}
```

La propiedad que en el caso del ejemplo anterior sería "color" especifica qué aspecto cambiaremos. En el ejemplo, cambiaríamos el color. Las propiedades que se quieren modificar para un mismo selector se pueden agrupar separándolas mediante un punto y coma.

```
p {text-align:center; color:red;}
```

Normalmente, describiremos una propiedad por línea de la manera siguiente:

```
h1 {
padding-left: 11em;
font-family: Georgia, "Times New Roman", Times, Serif;
color: red;
background-color: #d8da3d;
```

```
}
```

El valor, representado a la derecha de los dos puntos (:), establece el valor de la propiedad. Es importante recordar que si el valor está formado por más de una palabra, se tiene que poner entre comillas.

```
p {font-family: "sans serif";}
```

Hay diferentes tipos de selectores.

Los **selectores de elemento o etiqueta**, que hacen referencia directa a elementos o etiquetas que nos encontramos en el documento XHTML y son los que hemos visto en los ejemplos anteriores.

Los **selectores de id**, que hacen referencia al identificador de una sección o elemento dentro de nuestro XHTML.

```
#id {la declaración }
```

Nos permite aplicar estilo a una parte única de nuestro documento identificada por un atributo id, cuyo valor será el nombre del selector.

Hemos de tener en cuenta que el ID será un identificador único para un elemento dentro del documento, por lo que los estilos sólo podrán afectar a una única parte del documento, al contrario de los selectores de clase, que veremos más adelante.

Por ejemplo, si en nuestro documento XHTML tenemos un elemento

```
<div id="capsa"></div>
```

Para referirnos a él desde la hoja de estilos, escribiríamos lo siguiente dentro del documento CSS:

```
#capsa {font-family: "sans serif"; }
```

Y estos estilos sólo afectarían al contenido dentro del elemento que tiene el id=caja.

Los **selectores de clase** son los que se relacionan con los elementos o partes de nuestro XHTML, que tienen como valor del atributo clase el nombre del selector.

```
.clase1 {la declaración }
```

A diferencia de los selectores de id, los selectores de clase se pueden relacionar con diferentes elementos o partes de nuestro documento XHTML que estén identificados con la "clase" en la cual hemos definido los estilos.

Por ejemplo, si tenemos el siguiente código en el XHTML:

```
<p class="vermell">
Hola este texto es rojo. Pero la siguiente palabra es
<span class="amarilla">amarilla</span>
</p>
```

Para definir las clases "rojo" y "amarilla", que después podremos reutilizar en otras partes del documento, haríamos lo siguiente en nuestro documento CSS:

```
. rojo {color: #FF0000;}
. amarilla {color: #FFFF00; }
```

También tenemos los selectores contextuales, que se relacionan con elementos o partes de nuestro XHTML para el tipo de elemento y el contexto que se especifica en el selector.

Serán del tipo:

```
Contexto elemento {la declaración};
```

Cuando decimos contexto de un elemento nos referimos a los antecedentes del elemento, es decir, a los padres.

Por ejemplo, si tenemos el código siguiente:

```
<div id="cap">
<ul>
<li>elemento 1 lista</li>
<li>elemento 2 lista</li>
</ul>
</div>
```

El contexto del elemento 1 de la lista sería el siguiente "#cap ul li", que resumiendo querría decir "los li, dentro de ul, dentro de elemento con id cap". Por lo tanto, si en nuestra hoja de estilos tuviéramos lo siguiente:

```
#cap ul li {font-family: "sans serif";}
```

Estos estilos sólo se aplicarían a los elementos que cumplieran este contexto.

Hay que tener en cuenta que, para un elemento, siempre mandarán los selectores contextuales sobre tipos de selectores más genéricos.

También están los **selectores de atributo**, que se relacionan con los elementos que contienen un atributo con el valor que se especifica en el selector cuando se aplican los estilos. De este tipo selector no hablaremos mucho, ya que no lo soportan la mayoría de navegadores. Sólo pondremos la forma por si os lo encontráis alguna vez.

Son del estilo:

```
Selector[atributo="valor"] {la declaración };
```

Quiere decir que los estilos sólo se aplicarán al elemento dentro del documento que contenga el atributo que se especifica igual al valor del atributo que especificamos en el selector.

Como última cosa antes de empezar a ver las diferentes propiedades que podemos modificar para aplicar estilos en los documentos, cabe decir que tenemos que considerar que los estilos se heredan. ¿Qué quiere decir eso? Que algunos elementos heredan los estilos de su padre, a menos que se especifique lo contrario.

Ejemplo

Por ejemplo, si defino un tipo de letra para el elemento `<body>`, por defecto, si no especifico lo contrario, todos los elementos de mi documento dentro del *body*, tomarán ese tipo de letra.

2.3. Propiedades de texto: tipografía

Comencemos a ver las propiedades que podemos modificar en un selector que se aplique a texto: tipografía, color, fuente, decoración, alineación, espacios, etc.

De los elementos de texto, podemos modificar dos tipos de propiedades: las que se refieren a la **tipografía**, y las que se refieren al **espaciado**.

Las propiedades que podemos variar para la tipografía son las siguientes:

2.3.1. Color

Esta propiedad nos permite especificar el color del texto.

Valor: el valor puede ser el color en rgb (rgb (100%, 50%, 0%), rgb (255, 128, 0)), el color en hexadecimal (#ffa500, #008000, #800080), o uno de los 17 nombres de colores predefinidos (aqua, black, blue, fuchsia, gray, green, lime, maroon, navy, olive, orange, purple, red, silver, teal, white, yellow).

La propiedad color se hereda, o lo que es lo mismo, si no se especifica, toma el valor del padre.

Se puede aplicar a todos los elementos y lo soportan todos los navegadores.

Seguidamente, algunos ejemplos:

```
h2{ color: lime;}
p{ color: #ff0000; }
a{ color: rgb(255,255,0); }
```

Siguiendo con nuestro ejemplo, abrimos el archivo "estilos.css" y añadimos este código:

```
body {
color: #333333;
}

h2 {
color:#3366FF;
}
```

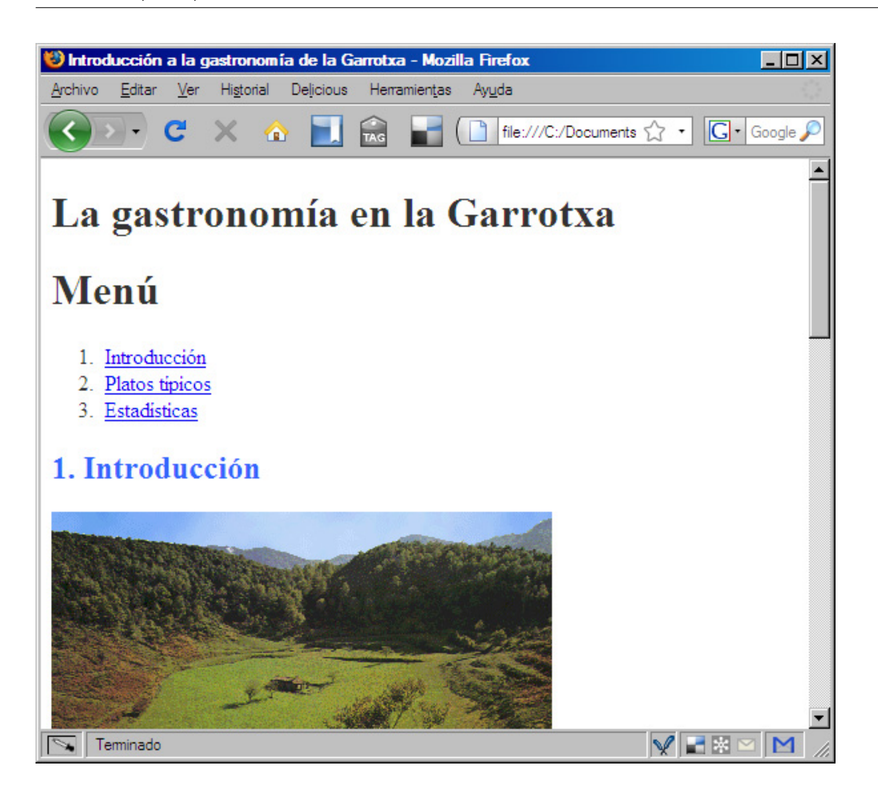
estils.css - Bloc de notas

Archivo Edición Formato Ver Ayuda

```
body {
 color: #333333;
}

h2 {
 color:#3366FF;
}
```

Vemos el efecto sobre la página que estamos construyendo. Si abrimos ahora el archivo "index.html" con nuestro navegador, veremos de qué manera afectan a estos estilos que acabamos de incorporar en el fichero estilos.css:



Los textos que están con encabezamientos de nivel 2, como "1. Introducción", se ven afectados por el estilo:

```
h2 {
color:#3366FF;
}
```

Veamos, pues, cómo el texto sale de color azul.

2.3.2. Font-family

Es una lista de nombres de familias de fuentes en orden de prioridad. En los ordenadores, no siempre tenemos todas las fuentes instaladas. Existen las llamadas "fuentes de sistema", que en principio las tiene todo el mundo. En esta propiedad se hará caso a la primera fuente de la lista que exista en el ordenador, donde se está visualizando el documento. Por ejemplo, si se especifica font-family: Arial, Verdana; y resulta que, en mi ordenador, no tengo Arial pero sí Verdana, lo vería en Verdana.

El valor de la propiedad, como hemos dicho, es una lista de familias de fuentes separadas por coma ",".

También es una propiedad que se hereda, se puede aplicar a todos los elementos y la soportan todos los navegadores principales.

Un ejemplo sería el siguiente:

```
p{
```

```
font-family: Arial, Helvetica, Sans-serif;
}
```

Seguimos con nuestro ejemplo "estilos.css". Añadiremos ahora este código:

```
body {
font-family: Georgia,"Times New Roman", Serif;
color: #333333;
}

h1 {
font-family: Helvetica, Geneva, Arial, Sans-serif;
}

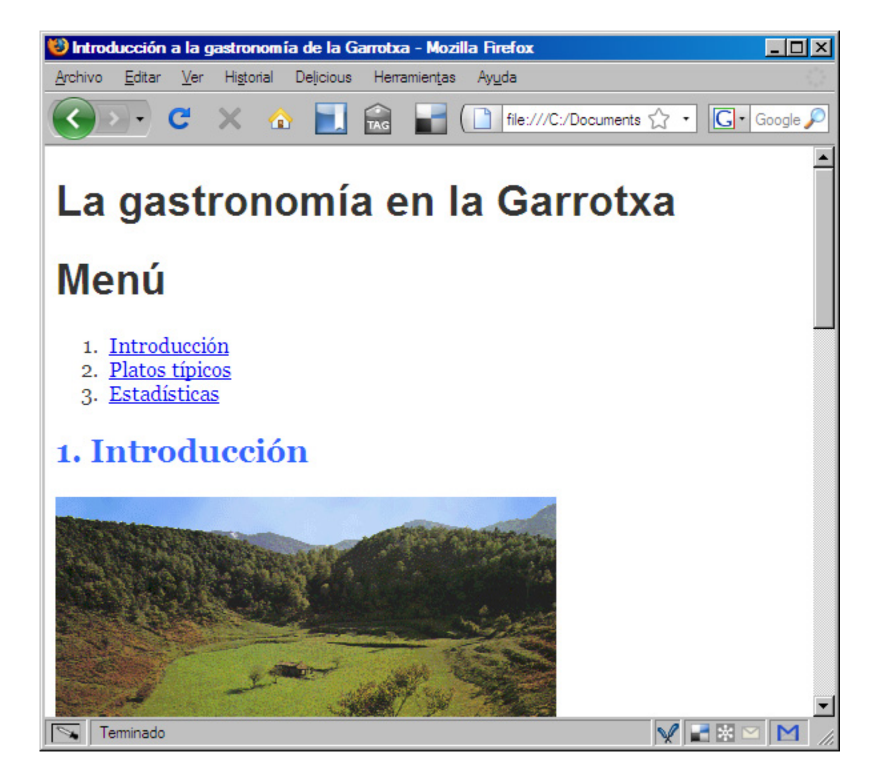
h2 {
color:#3366FF;
}
```

```
estils.css - Bloc de notas
Archivo Edición Formato Ver Ayuda
body {
 font-family: Georgia,"Times New Roman", serif;
 color: #333333;
}

h1 {
 font-family: Helvetica, Geneva, Arial, sans-serif;
}

h2 {
 color:#3366FF;
}
```

Visualizamos el archivo "index.html" en un navegador y vemos los resultados:



2.3.3. Font-size

Esta propiedad nos permite establecer la medida de la fuente a utilizar. Su valor se puede especificar con una "medida", un "porcentaje" o alguna de las medidas absolutas o relativas predefinidas.

Lo que explicaremos ahora sobre las diferentes maneras de asignar las medidas se puede aplicar también a otras propiedades que requieran como valor una medida.

Una "medida" será un número decimal seguido de una unidad.

Las **unidades absolutas** son las siguientes:

mm: (milímetro)
cm: (centímetro, 1 cm = 10 mm)
in: (pulgada (inch), 1in = 2,54 cm)
pt: (punto, 1 pt = 1/72 in)
pc: (pica, 1 pc = 12 pt)

Las **unidades relativas** son aquellas que, a pesar de la redundancia, son relativas a una medida del documento. En general, a la medida de la fuente. Estas unidades son:

em: (1 em = medida de la fuente actual)
ex: (1 ex = altura de la letra 'x' en la fuente actual)

Si la medida que especificamos es de 2 em, y la medida de la fuente actual es 10 px, la medida que hemos especificado será de 20 px. El hecho de especificar las medidas en unidades relativas es especialmente útil en el caso de hacer diseños que se adapten a la medida de la pantalla del usuario.

La unidad más utilizada, sin embargo, es el px (píxel).

También podemos especificar las medidas de forma relativa en porcentajes.

En el caso de las medidas predefinidas, también las tenemos de dos tipos:

1) Las **absolutas**,

Medidas absolutas

Las medidas absolutas con su correspondencia con los encabezamientos y el factor de escala:

Medida	Factor de escala	Encabezamiento
xx-large	2	h1
x-large	1.5	h2
Large	1.2	h3
Medium	1	h4
Small	0.89	h5
x-small	0.75	
xx-small	0.6	h6

2) Las **relativas** en las cuales podemos especificar "larger", que aumentará la fuente actual un paso y "smaller" que la decrementará.

El valor por defecto de la propiedad *font-size* es el valor predefinido "medium". Es decir, si no especificamos nada, el valor será "medium". Es una propiedad que hereda, lo que quiere decir que toma la medida de su padre si no se especifica lo contrario. Se puede aplicar a todos los elementos y la soportan todos los navegadores principales.

Aquí podemos ver algunos ejemplos:

```
p{
font-size:10px;
}

.classe1{
font-size:x-large;
}

#peu{
font-size:smaller;
}
```

En nuestro ejemplo, añadimos el siguiente código en el archivo "estilos.css":

```
body {
font-family: Georgia,"Times New Roman", Serif;
color: #333333;
font-size: 11px;
}

h1 {
font-family: Helvetica, Geneva, Arial, Sans-serif;
}
```

```
h2 {
color:#3366FF;
}
```

estils.css - Bloc de notas

Archivo Edición Formato Ver Ayuda

```
body {
 font-family: Georgia,"Times New Roman", serif;
 color: #333333;
 font-size: 11px;
}

h1 {
 font-family: Helvetica, Geneva, Arial, sans-serif;
}

h2 {
 color:#3366FF;
```

Ahora veremos el impacto de éste cuando visualizamos nuestro archivo en el navegador:



Como vemos, la tipografía de las listas que hay dentro del "body" ha tomado la nueva medida, igual que lo ha hecho el resto de texto genérico.

2.3.4. Font-style

Especifica el estilo oblicuo o itálico dentro de la familia de fuente actual.

Los valores que puede tomar esta propiedad son: "normal", "italic", u "oblique".

El valor por defecto es "normal"; es una propiedad que hereda del padre el valor si no se especifica, es aplicable a todos los elementos y la soportan todos los navegadores.

```
.clase1{font-style:oblique;}

#cos{font-style:italic;}
```

2.3.5. Font-variant

Esta propiedad determina si la fuente se muestra en mayúsculas de tipo normal o pequeñas "small-caps". Éstas se muestran de forma que todas las letras de una determinada palabra están en mayúsculas con caracteres ligeramente mayores que las minúsculas.

Los valores que se pueden especificar son "normal" o "small-caps". El valor por defecto es "normal"; es una propiedad que hereda del padre el valor si no se especifica, es aplicable a todos los elementos y la soportan todos los navegadores.

2.3.6. Font-weight

Especifica el grueso de la fuente actual. Podemos definir que sea en negrita, normal o delgada.

Los valores que se pueden especificar son: "normal", "bold", "lighter", o uno de 9 valores numéricos (100, 200..., 900). El valor por defecto es "normal". Es una propiedad que hereda del padre el valor si no se especifica, es aplicable a todos los elementos y la soportan todos los navegadores.

```
p{ font-weight:bold; }
 a{ font-weight:400: }
```

2.3.7. Text-decoration

Especifica si el texto aparece subrayado, con una línea superior, rayado, o con parpadeo.

Los valores que se pueden especificar son: "none" (ninguno), "underline" (subrayado), "overline"(línea superior), line-through" (rallado) o "blink" (parpadeo).

El valor por defecto es "none"; es una propiedad que no hereda del padre el valor si no se especifica, es aplicable a todos los elementos y la soportan todos los navegadores.

A continuación, podemos ver el ejemplo más típico con los enlaces. Los enlaces están, por defecto, subrayados y de color azul. Una cosa que no hemos dicho es que los enlaces pueden tener diferentes estados. El estado normal, el estado cuando tenemos el ratón encima (mouseover), o el estado de visitado, que quiere decir que ya le hemos hecho clic.

En el ejemplo siguiente, hacemos referencia al estado normal "a", y al estado de mouseover "a:hover". Por lo tanto, el enlace se verá sin subrayado en el estado normal, y cuando nos pongamos encima (mouseover) aparecerá la línea superior tal como estamos especificando.

```
a{ text-decoration:none; }

a:hover{ text-decoration:overline; }
```

2.3.8. Text-transform

Permite al texto transformarse en alguna de las cuatro propiedades: "capitalize" (la primera letra con mayúscula), "uppercase" (todas las letras con mayúsculas), "lowercase" (todas las letras con minúsculas), "none" (ningún efecto sobre el texto).

Los valores que se pueden especificar son: "capitalize", "uppercase", "lowercase", "none". El valor por defecto es "none"; es una propiedad que hereda del padre el valor si no se especifica lo contrario, es aplicable a todos los elementos y la soportan todos los navegadores.

```
p{ text-transform:lowercase; }
```

2.4. Propiedades de texto: espaciado

Las propiedades que podemos variar con respecto al espaciado de texto son las siguientes:

2.4.1. White-space

Especifica cómo se comportarán las tabulaciones, los saltos de línea y el espacio extra en blanco en el contenido del elemento especificado.

Los valores que se pueden especificar son: "normal","pre", "pre-wrap", "pre-line". El valor por defecto es "normal"; es una propiedad que hereda del padre el valor si no se especifica, es aplicable a elementos de blog y la soportan todos los navegadores, excepto las propiedades "pre-wrap" y "pre-line", que fue añadido en el CSS 2.1 y sólo la soportan ciertos navegadores.

```
p{ white-space:nowrap; }
```

2.4.2. Text-align

Especifica la alineación horizontal de las líneas de texto. Podremos alinear un texto a la izquierda o a la derecha, en el centro o justificado.

Los valores que se pueden especificar son: "left", "right", "center", "justify". El valor por defecto es "none", es una propiedad que hereda del padre el valor si no se especifica, es aplicable a todos los elementos de blog y la soportan todos los navegadores.

```
p{ text-align:justify; }
```

2.4.3. Text-indent

Especifica la imbricación de la primera línea de texto de un párrafo.

El valor es una medida. El valor por defecto es "0", es una propiedad que hereda del padre el valor si no se especifica, es aplicable a todos los elementos de blog, y la soportan todos los navegadores.

```
p{ text-indent:10px; }
```

2.4.4. Line-height

Especifica la distancia mínima entre líneas base de texto.

Los valores que se pueden especificar son un factor, una medida o bien un porcentaje. El valor por defecto depende del navegador, es una propiedad que hereda del padre, el valor si no se especifica es aplicable a todos los elementos, y la soportan todos los navegadores.

El factor es un número decimal (por ejemplo, 1.2) que multiplicará la medida de fuente actual para calcular la nueva medida.

```
p{ line-height:1.2; }
```

2.4.5. Word-spacing

Especifica la cantidad adicional de espacio entre palabras.

Los valores que se pueden especificar son "normal" o una medida. El valor por defecto es "normal"; es una propiedad que hereda del padre, el valor si no se especifica lo contrario es aplicable a todos los elementos y la soportan todos los navegadores.

```
p{ word-spacing:10; }
```

2.4.6. Letter-spacing

Especifica la cantidad adicional de espacio entre letras.

Los valores que se pueden especificar son "normal" o una medida. El valor por defecto es "normal", es una propiedad que hereda del padre, el valor si no se especifica, es aplicable a todos los elementos y la soportan todos los navegadores.

```
p{ letter-spacing:10; }
```

2.4.7. Vertical-align

Sirve para alterar la ubicación vertical de un elemento en línea, en relación a su elemento padre o a la línea del elemento.

El valor puede ser un porcentaje relativo a la altura del elemento, que elevará la línea de base del elemento en la cantidad especificada por encima de la línea de base del padre. No se permiten valores negativos.

El valor también puede ser una palabra clave. Las siguientes palabras clave afectan a la ubicación en relación con el elemento padre:

- **Baseline**: alinea líneas base del elemento y el padre;
- **Middle**: alinea el punto medio vertical del elemento con la línea de base más la mitad de la altura de la letra "x" del padre.
- **Sub**: subíndice.
- **Super**: superíndice.
- **Text-top**: alinea las partes inferiores del elemento y la fuente del elemento padre.
- **Text-bottom**: alinea las partes inferiores del elemento y la fuente del elemento padre.

El valor por defecto es "baseline"; es una propiedad que no hereda del padre, el valor si no se especifica es aplicable a todos los elementos de línea, y la soportan todos los navegadores.

```
img{ vertical-align:middle; }
```

2.4.8. Direction

Especifica la dirección de escritura. Ésta puede ser de izquierda a derecha ("ltr") o de derecha a izquierda ("rtl"). Se utiliza, básicamente, para los lenguajes que se escriben de derecha a izquierda como el árabe.

El valor por defecto es "ltr", es una propiedad que hereda del padre el valor si no se especifica, es aplicable a todos los elementos y la soportan todos los navegadores.

```
p{ direction:rtl; }
```

2.5. Propiedades de disposición: estructura de caja

En cuanto en las propiedades de disposición ("layout", estructura de cajas, altura, anchura, margen, "padding", fondo, etc.), las podemos dividir en tres tipos:

1) Las de "estructura de caja",
2) las de posicionamiento,
3) y las de fondo.

Las propiedades de "estructura de caja" son las siguientes:

2.5.1. Margin

Especifica las cuatro propiedades individuales del margen de una vez.

```
div {margin: 1em 2em 3em 4em }
```

Donde cada uno de los valores corresponde a los márgenes superior, derecho, inferior e izquierdo respectivamente en este orden. Si sólo tenemos un valor, se aplica a todos los lados. Si tenemos dos o tres valores, los valores restantes se toman del lado opuesto.

```
margin: 1em;
```

(todos los márgenes = 1em)

```
margin: 1em 2em;
```

(superior e inferior = 1em, derecho e izquierdo = 2em)

```
margin: 1em 2 m 3m;
```

(superior = 1em, derecho e izquierdo = 2em, inferior = 3em)

```
margin: 1em 2em 3em 4em;
```

(superior = 1em, derecho = 2em, inferior = 3em, izquierdo = 4em)

Los posibles valores son hasta cuatro valores que pueden ser una medida, un porcentaje o "auto". El valor por defecto es "0", es una propiedad que no hereda del padre el valor si no se especifica, es aplicable a todos los elementos y la soportan todos los navegadores.

2.5.2. Margin-top, margin-right, margin-bottom, margin-left

Son cuatro propiedades y especifican el espacio entre el elemento seleccionado y los elementos que están al lado. Es decir, los márgenes, igual que hacía la propiedad "margin", pero especificando cada uno por separado.

Los valores que se pueden especificar son: un número, un porcentaje o bien "auto". El valor por defecto es "0", es una propiedad que no hereda del padre el valor si no se especifica, es aplicable a todos los elementos y la soportan todos los navegadores.

```
p{
margin-bottom:10px;
margin-left:auto;
margin-right:auto;
margin-top:20px;
}
```

Seguiremos ahora con nuestro archivo de ejemplo. Abriremos el archivo "estilos.css" y añadiremos el código siguiente:

```
body {
font-family: Georgia,"Times New Roman", Serif;
color: #333333;
font-size: 11px;
}

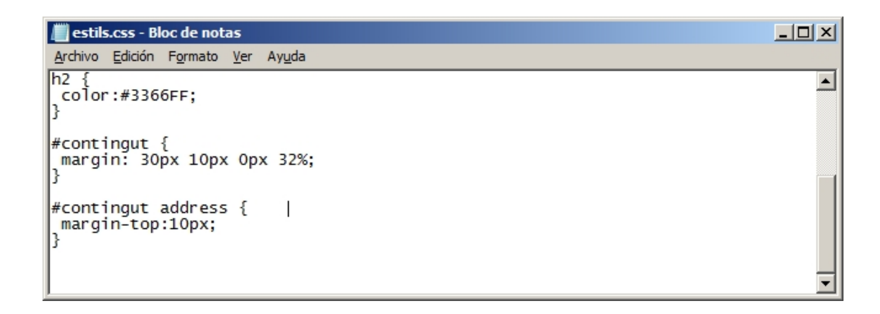
h1 {
font-family: Helvetica, Geneva, Arial, Sans-serif;
}

h2 {
color:#3366FF;
}

#contenido {
margin: 30px 10px 0px 32%;
}

#contenido address {
margin-top:10px;
```

```
    }
```



Vemos la visualización en el navegador:

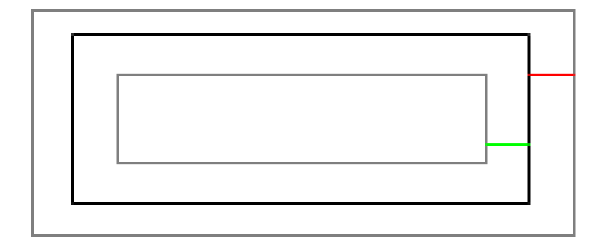


Podemos observar que todo el contenido dentro de la capa "content" se ha desplazado a la derecha la cantidad que le hemos especificado.

2.5.3. Padding

Para que nos entendamos, el "padding" es el "relleno". Si tenemos una caja que tiene contenido, el "margin" (margen), como hemos visto, sería el espacio que esta caja mantendrá en la parte de fuera enfrente de contenidos vecinos. Es decir, si especificamos 10px, nuestra caja estará separada del contenido 10px (dejamos 10px de margen). En cambio, el "padding" (relleno) es el espacio que tendrá que respetar el contenido de dentro de la caja con los límites de la caja.

En el siguiente esquema, lo veremos más claro. La línea verde sería el "padding" y la línea roja el "margin".



Después de esta nota, seguimos con el "padding". El "padding" nos especifica las cuatro propiedades de espacio entre el borde y el contenido del texto de una vez. Se aplica siguiendo el mismo mecanismo que la propiedad "margin".

Los posibles valores son hasta cuatro valores, que pueden ser una medida o un porcentaje. El valor por defecto es "0". Es una propiedad que no hereda del padre el valor si no se especifica, es aplicable a elementos y la soportan todos los navegadores.

2.5.4. Padding-top, padding-right, padding-bottom, padding-left

Especifica la cantidad de espacio que hay que insertar entre el contenido de un elemento y su margen o borde.

Los valores que se pueden especificar son: un número o un porcentaje. El valor por defecto es "0". Es una propiedad que no hereda del padre el valor si no se especifica, es aplicable a elementos y la soportan todos los navegadores.

```
p{
padding-bottom:10px;
padding-left:auto;
padding-right:auto;
padding-top:20px;
}
```

Añadiremos algunos "paddings"en nuestro archivo "estilos.css":

```
body {
padding-left: 5px;
font-family: Georgia,"Times New Roman", Serif;
color: #333333;
font-size: 11px;
}

h1 {
font-family: Helvetica, Geneva, Arial, Sans-serif;
}

h2 {
color:#3366FF;
```

```
}

#cabecera {
padding-left:5px;
}



#menu {
padding-top:5px;
padding-bottom:5px;
padding-left: 5px;
}

#contenido {
margin: 30px 10px 0px 32%;
padding-left:5px;

}
#contenido address {
margin-top:10px;
}

#contenido img {
padding-right:10px;
padding-bottom:5px;
}
```

estils.css - Bloc de notas

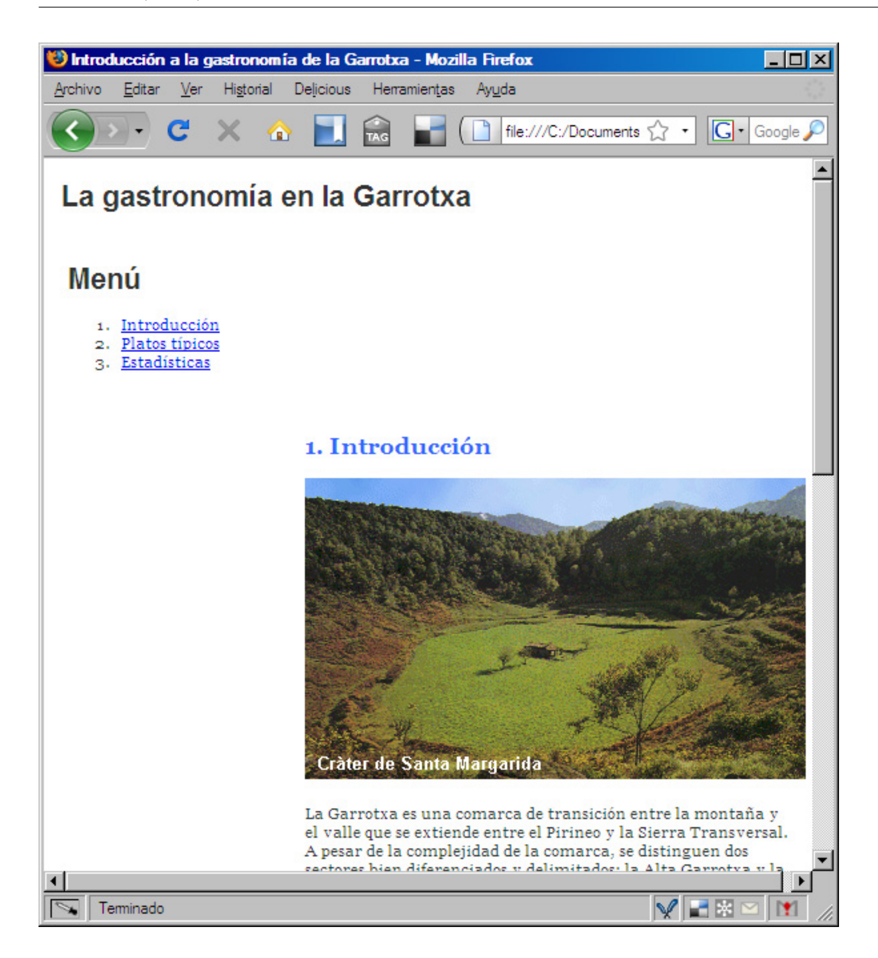
Archivo Edición Formato Ver Ayuda

```
 margin: 30px 10px 0px 32%;
 padding-left:5px;
}


#contingut address {
 margin-top:10px;
}

#contingut img {
 padding-right:10px;
 padding-bottom:5px;
}
```

Y la visualización del archivo "index.html" en el navegador será así:



2.5.5. Border, border-top, border-right, border-bottom, border-left

Cada una especifica la anchura, el color y el estilo de cada uno de los bordes, a excepción del "border" que lo aplica a todos los lados en general.

Los valores que se pueden especificar están separados por un espacio: anchura (en píxeles o en porcentaje), el color (en RGB) y estilo. Son propiedades que no heredan el valor si no se especifica, son aplicables elementos y las soportan todos los navegadores.

Los valores que podemos especificar en el estilo del borde son:

- **None**: ningún estilo.
- **Dotted**: línea de puntos.
- **Dashed**: línea de barras.
- **Solid**: línea sólida.
- **Double**: doble grueso de línea.
- **Groove**, **ridge**, **inset** y **outset**: diferentes estilos de bordes en 3D.

A continuación, podemos ver algunos ejemplos:

```
p{
border: 1px solid #FFFFFF;
```

```
}

. caja {
border-bottom: 1px dashed #FF0000;
border-top: 4px groove #00FF00;
border-left:1px double #00FF00;
border-right: 2px solid #0000FF;
}
```

Añadiremos el siguiente código en el .css para conseguir que nuestra tabla tenga un pequeño filete:

```
body {
padding-left: 5px;
font-family: Georgia,"Times New Roman", Serif;
color: #333333;
font-size: 11px;
}

h1 {
font-family: Helvetica, Geneva, Arial, Sans-serif;
}

h2 {
color:#3366FF;
}

#cabecera {
padding-left:5px;
}

#menu {
padding-top:5px;
padding-bottom:5px;
padding-left: 5px;
}

#contenido {
margin: 30px 10px 0px 32%;
padding-left:5px;
}

#contenido table {
border:1px solid #3366FF;
}

#contenido address {
```

```
    margin-top:10px;
    }

    #contenido img {
    padding-right:10px;
    padding-bottom:5px;
    }
```

```
estils.css - Bloc de notas
Archivo Edición Formato Ver Ayuda
 padding-left:5px;
}

#contingut table {
 border:1px solid #3366FF;
 border-spacing:10px 10px;
}

#contingut address {
 margin-top:10px;
}

#contingut img {
```

Vemos cómo se visualiza en nuestro navegador:



Hemos añadido un filete de 1 píxel de grueso, de estilo "sólido" y de color azul.

2.5.6. Border-width

Especifica el grueso del borde. Este valor tiene que ser bastante grande para apreciarlo correctamente. Podemos especificar a la vez las cuatro propiedades del grueso del borde. Cada uno de los valores corresponde al grueso del borde superior, derecho, inferior e izquierdo, respectivamente. Si sólo tenemos un valor, se aplica a todos los lados. Si tenemos dos o tres valores, los valores restantes se toman del lado opuesto, tal como hemos visto con el "margin" o el "padding".

Los valores que se pueden especificar son: una de las tres palabras clave ("thin", "médium" o "tic") o un número entre el 1 y el 4. Es una propiedad que no hereda del padre el valor si no se especifica, es aplicable a elementos y la soportan todos los navegadores.

```
p{
border-width:thin;
}
```

2.5.7. Border-top-width, border-right-width, border-bottom-width, border-left-width

Especifican la anchura del borde superior, lateral derecho, inferior o lateral izquierdo de un elemento.

Los valores que se pueden especificar son: una de las tres palabras clave ("thin", "médium" o "thick") o un número entre el 1 y el 4. Es una propiedad que no hereda del padre el valor si no se especifica, es aplicable a todos los elementos y la soportan todos los navegadores.

```
.caixa {
  border-bottom-width: 1px;
  border-top-width: 4px;
  border-left-width: 1px;
  border-right-width: 2px;
  }
```

2.5.8. Border-style

Especifica el estilo de hasta los cuatro bordes de la caja de un elemento a la vez.

Los valores que se pueden especificar son una de las siguientes palabras clave: none, dotted, dashed, solid, double, groove, ridge, inset, outset, hidden. Es una propiedad que no hereda del padre el valor si no se especifica, es aplicable a elementos y la soportan todos los navegadores.

```
p{
  border-style:double;
  }
```

2.5.9. Border-top-style, border-right-style, border-bottom-style, border-left-style

Especifica el estilo del borde de un lado específico del elemento.

Los valores que se pueden especificar son una de las siguientes palabras clave: none, dotted, dashed, solid, double, groove, ridge, inset, outset, hidden. Es una propiedad que no hereda del padre el valor si no se especifica, es aplicable a todos los elementos y la soportan todos los navegadores.

```
.caixa {
  border-bottom-style: dotted;
  border-top-style: groove;
  border-left-style: dashed;
  border-right-style: solid;
  }
```

2.5.10. Border-color

Especifica el color de hasta los 4 bordes de la caja de un elemento a la vez.

Los valores que se pueden especificar pueden ser: un valor rgb, un valor hexadecimal, "transparente" o uno de los 17 nombres de color predeterminados que hemos visto antes en la propiedad "color". Por defecto, coge la propiedad de color del elemento. Es una propiedad que no hereda del padre el valor si no se especifica, es aplicable a todos los elementos y la soportan todos los navegadores.

```
p{
  border-color: #000000;
  }
```

2.5.11. Border-top-color, border-right-color, border-bottom-color, border-left-color

Especifica el color de un borde concreto de la caja del elemento.

Los valores que se pueden especificar son: un valor rgb, un valor hexadecimal, "transparente" o uno de los 17 nombres de color predeterminados que hemos visto antes en la propiedad "color". Por defecto coge la propiedad de color del elemento. Es una propiedad que no hereda del padre el valor si no se especifica, es aplicable a todos los elementos y la soportan todos los navegadores.

```
.caixa {
  border-bottom-color: #CC9933;
  border-top-color: #00FF99;
  border-left-color: #999900 ;
  border-right-color: #33CCCC;
  }
```

2.5.12. Width

Especifica la anchura del área de contenido de un elemento.

Los valores que se pueden especificar pueden ser: uno, un porcentaje, o "auto". Es una propiedad que no hereda, es aplicable a todos los elementos de blog y elementos de sustitución, y la soportan todos los navegadores.

```
#content{
  width: 100%;
  }
```

A continuación, especificaremos una anchura en la parte del menú de nuestro ejemplo:

```
body {
padding-left: 5px;
font-family: Georgia,"Times New Roman", Serif;
color: #333333;
font-size: 11px;
}

h1 {
font-family: Helvetica, Geneva, Arial, Sans-serif;
}

h2 {
color:#3366FF;
}

#cabecera {
padding-left:5px;
}

#menu {
width: 30%;
background-color:#e5e5e5;
padding-top:5px;
padding-bottom:5px;
padding-left: 5px;
}

#contenido {
margin: 30px 10px 0px 32%;
padding-left:5px;
}
```

```
#contenido table {
border:1px solid #3366FF;
}

#contenido address {
margin-top:10px;
}

#contenido img {
padding-right:10px;
padding-bottom:5px;
}
```

estils.css - Bloc de notas

Archivo Edición Formato Ver Ayuda

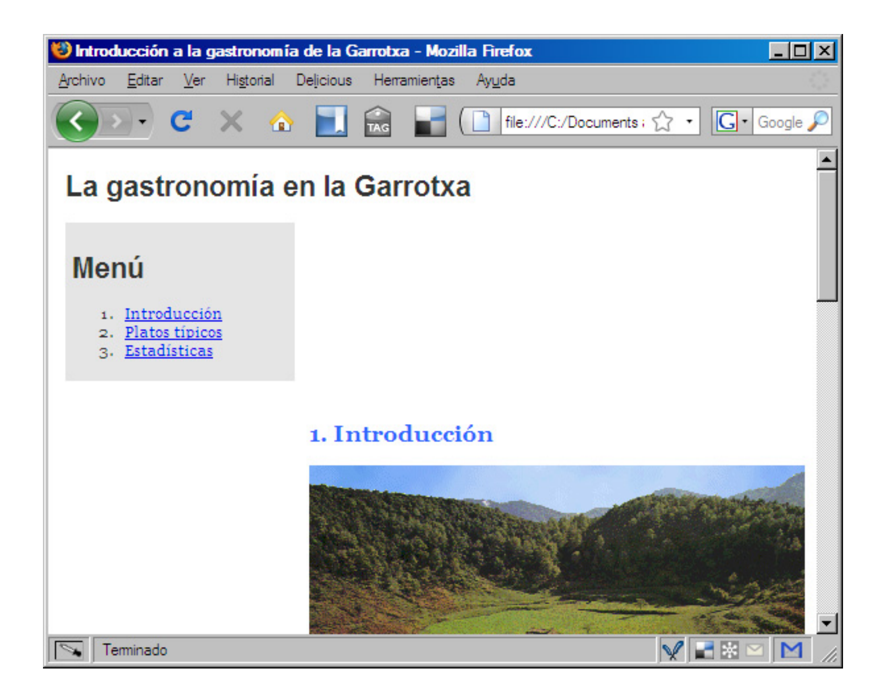
```
#capçalera {
 padding-left:5px;
}

#menu {
 width: 30%;
 background-color:#e5e5e5;
 padding-top:5px;
 padding-bottom:5px;
 padding-left: 5px;
}

#contingut {
```

Para poder visualizar cómo afecta el hecho de haber asignado una medida relativa del 30% a nuestro menú, hemos añadido un color de fondo al menú.

Veremos cómo se visualiza en el navegador este código:



Para comprobar el efecto de haber asignado una medida relativa al menú, podemos estirar y encoger la ventana de nuestro navegador. De esta manera, podremos ver cómo el contenido se ajusta a la medida especificada:



2.5.13. Min-width

Protege el elemento de ser demasiado estrecho.

Los valores que se pueden especificar pueden ser: una medida o un porcentaje. Por defecto es "0". Es una propiedad que no hereda del padre el valor si no se especifica, es aplicable a todos los elementos excepto los elementos de no sustitución y los elementos de tabla, y la soportan todos los navegadores excepto el Internet Explorer.

```
#content{
  min-width: 25px;
  }
```

2.5.14. Max-width

Protege el elemento de ser demasiado amplio.

Los valores que se pueden especificar pueden ser: una medida, un porcentaje o "none". Por defecto es "none". Es una propiedad que no hereda del padre el valor si no se especifica, es aplicable a todos los elementos, excepto a los elementos de no sustitución y a los elementos de tabla, y la soportan todos los navegadores excepto Internet Explorer.

```
#content{
  max-width: 100%;
  }
```

2.5.15. Height

Especifica la altura del área de contenido del elemento.

Los valores que se pueden especificar pueden ser: una medida, o "auto". Por defecto es "auto". Es una propiedad que no hereda del padre el valor si no se especifica, es aplicable a todos los elementos de blog y los elementos de sustitución, y la soportan todos los navegadores excepto Internet Explorer.

```
#content{
  height: 100%;
  }
```

2.5.16. Min-height

Protege el elemento de ser demasiado pequeño en altura.

Los valores que se pueden especificar pueden ser: una medida o un porcentaje. Por defecto es "0". Es una propiedad que no hereda del padre el valor si no se especifica, es aplicable a todos los elementos excepto los elementos de no sustitución y los elementos de tabla, y la soportan todos los navegadores excepto Internet Explorer.

```
#content{
  min-height: 20px;
  }
```

2.5.17. Max-height

Protege el elemento de ser demasiado alto.

Los valores que se pueden especificar pueden ser: una medida, o "auto". Por defecto es "auto". Es una propiedad que no hereda del padre el valor si no se especifica, es aplicable a todos los elementos de blog y los elementos de sustitución, y la soportan todos los navegadores excepto Internet Explorer.

```
#content{
  max-height: 100%;
  }
```

2.5.18. Overflox

Determina la manera de visualizar los elementos hijos de un elemento que no caben en el área de contenido de su padre.

Los **valores** pueden ser:

- **Visible**: si el contenido del elemento excede las dimensiones se mostrará igualmente. El comportamiento puede ser variable dependiendo del navegador.

- **Hidden**: el elemento se mostrará correctamente, mientras que el contenido que exceda de las dimensiones no será visible.

- **Auto**: si el contenido excede las dimensiones del padre o contenedor, aparecerán las barras de "scroll" para posibilitar la lectura.

- **Scroll**: siempre aparecen las barras de "scroll". En caso de que el contenido no exceda las dimensiones, éstas quedarán deshabilitadas. En el momento en que el contenido exceda las dimensiones, se habilitarán.

Por defecto, es "visible". Es una propiedad que no hereda del padre el valor si no se especifica, es aplicable a todos los elementos de blog y los elementos de sustitución, y la soportan todos los navegadores.

```
#content{
  overflow: auto;
  }
```

2.5.19. Clip

Define qué parte del contenido será visible.

Los valores que se pueden especificar pueden ser: una forma, o "auto". La forma se define de la siguiente manera:

```
rect(5px, 110px, 55px, 10px)
```

Donde los valores corresponden a superior, derecha, inferior e izquierda. Este ejemplo se refiere a un rectángulo que empieza 5px desde la parte superior y 10px desde la izquierda y tiene 100px de anchura y 50px de altura.

Por defecto, es "auto". Es una propiedad que no hereda del padre el valor si no se especifica, es aplicable a todos los elementos con posición absoluta, y la soportan todos los navegadores excepto Internet Explorer.

2.6. Propiedades de disposición: posicionamiento

Las propiedades que nos permitirán modificar el posicionamiento de los elementos son las siguientes:

2.6.1. Display

Determina cómo se mostrará un elemento. Los valores que podemos especificar son:

- **Blog**: este valor provoca que un elemento genere una caja de blog principal, como si hubiera un salto de línea antes y después del elemento.

- **Inline**: este valor provoca que un elemento genere unas o más cajas de línea, es decir, que los elementos se mostrarán uno tras otro en la misma línea mientras quepan allí.

- **List-item**: este valor provoca que un elemento genere una caja de blog principal y una caja de línea de tipo "list-item".

- **Marker**: este valor declara que el contenido generado antes o después de una caja será un marcador.

- **None**: Este valor provoca que un elemento no genere ninguna caja (es decir, el elemento no tiene ningún efecto sobre la composición) y los elementos descendentes tampoco generen cajas. Se tiene que destacar que este valor no crea una caja invisible, sino que hace que el elemento desaparezca del todo. Esta es la diferencia hacia las propiedades sobre visibilidad, que provocan que un elemento pueda ser invisible pero siga ocupando un espacio en la página.

- **Run-in y compact**: estos valores crean cajas de blog o de línea según el contexto y tienen un comportamiento como el que conocemos para las listas de definición.

- **Table**, **inline-table**, **table-row-group**, **table-column**, **table-column-group**, **table-header-group**, **table-footer-group**, **table-row**, **table- cell y table-capt**: Estos valores provocan que un elemento se comporte como un elemento tabla.

El valor por defecto de la propiedad display es "inline". Es una propiedad que no hereda del padre el valor si no se especifica, es aplicable a todos los elementos con posición absoluta, y la soportan todos los navegadores.

```
#contenido {
  display: blog;
  }
```

2.6.2. Position

Especifica el método por el cual se determina la posición de la caja del elemento. Los elementos que no son "estáticos" utilizan las propiedades de posicionamiento "top, right, bottom, y left" que veremos más adelante.

Los **valores** que podemos especificar son:

- **Static**: con este valor, la caja se sitúa dentro del flujo normal de la página.

- **Relative**: la posición de la caja se ajusta en relación con su posición normal dentro de la página. Cuando una caja se posiciona relativamente, la caja siguiente se sitúa como si aquélla no se hubiera desplazado.

- **Absolute**: las cajas son sacadas de su flujo natural de página y su posición se especifica con las propiedades "left", "right", "top","bottom". Estas propiedades especifican los desplazamientos con respecto al blog padre, por lo cual los elementos posicionados absolutamente no tienen ninguna influencia en las cajas siguientes.

- **Fixed**: el posicionamiento "fixed" es una subcategoría del posicionamiento absoluto. La única diferencia es que, para una caja posicionada fija, el blog padre es establecido por el acceso visual (la pantalla del monitor) y el elemento no se mueve cuando se realiza un desplazamiento. Eso quiere decir que cuando se hace scroll en la página, estos elementos mantienen su posición fija.

El valor por defecto es "static". Es una propiedad que no hereda del padre el valor si no se especifica, es aplicable a elementos con posición absoluta y la soportan todos los navegadores.

```
#contenido {
  position:absolute;
  top:20px;
  left:20px;
  }
```

2.6.3. Top, right, bottom, left

Especifica la posición de un elemento fijo, absoluto, o posicionado de forma relativa.

Los valores que se pueden especificar son: una medida, un porcentaje o "auto". Por defecto, es "auto". Es una propiedad que no hereda del padre el valor si no se especifica, es aplicable a elementos con una posición diferente a estática y la soportan todos los navegadores.

En el ejemplo de la propiedad "position" anterior el elemento con id contenido de nuestro documento se colocaría absolutamente en 20px de separación de la parte superior y de la parte izquierda.

2.6.4. Float

Mueve un elemento a derecha o izquierda del elemento del que es hijo hasta que no encuentra otro elemento de blog de nivel. Los elementos flotados son eliminados del flujo normal de la página.

Los valores que se pueden especificar son: "left", "right" o "none". Por defecto, es "none". Es una propiedad que no hereda del padre el valor si no se especifica, es aplicable a todos los elementos y la soportan todos los navegadores.

```
#contenido {
  float: left;
  }
```

Nuestro siguiente objetivo es hacer que el menú quede a la izquierda del contenido; por lo tanto, añadiremos un "float:left" a nuestro código:

```
body {
padding-left: 5px;
font-family: Georgia,"Times New Roman", Serif;
color: #333333;
font-size: 11px;
}

h1 {
font-family: Helvetica, Geneva, Arial, Sans-serif;
}

h2 {
color:#3366FF;
}

#cabecera {
padding-left:5px;
}

#menu {
float: left;
width: 30%;
background-color:#e5e5e5;
padding-top:5px;
padding-bottom:5px;
padding-left: 5px;
```

```
}

#contenido {
margin: 30px 10px 0px 32%;
padding-left:5px;
}

#contenido table {
border:1px solid #3366FF;
}

#contenido address {
margin-top:10px;
}

#contenido img {
float:left;
padding-right:10px;
padding-bottom:5px;
}
```

estils.css - Bloc de notas

Archivo Edición Formato Ver Ayuda

```
#contingut table {
 border:1px solid #3366FF;
}

#contingut address {
 margin-top:10px;
}

#contingut img {
 float:left;
 padding-right:10px;
 padding-bottom:5px;
}
```

Vemos ahora cómo se visualiza el código en el navegador:



2.6.5. Clear

Esta propiedad indica cuáles de los lados de la caja de un elemento no pueden quedar adyacentes a una caja flotante anterior. Esta propiedad sólo se puede especificar para elementos de flujo de blog (incluyendo también los elementos flotantes).

Los valores tienen los siguientes **significados**:

- **Left**: el margen superior de la caja generada se aumenta bastante para que su parte superior quede justo debajo de la parte inferior de cualquier caja flotante a la izquierda que aparezca antes en el documento.

- **Right**: el margen superior de la caja generada se aumenta bastante para que su parte superior quede justo debajo de la parte inferior de cualquier caja flotante a la derecha que aparezca antes en el documento.

- **Both**: la caja generada se mueve debajo de todas las cajas flotantes que aparezcan antes en el documento.

- **None**: no existe ninguna restricción a la posición de la caja del elemento respecto de los elementos flotantes.

Por defecto, es "none". Es una propiedad que no hereda del padre el valor si no se especifica, es aplicable a todos los elementos y la soportan todos los navegadores.

```
#contenido {
  clear: both;
```

```
    }
```

2.6.6. Z-index

Especifica en qué orden de superposición se apilan los elementos unos sobre los otros. Nos dice el nivel de profundidad en que se encuentra cada elemento. Normalmente, los últimos elementos que aparecen en el código del documento se apilan sobre los elementos que aparecen antes. Cuanto mayor sea el "z-index", más por debajo de él estará.

Los valores que se pueden especificar pueden ser: un número entero, o "auto". Por defecto, es "auto". Es una propiedad que no hereda del padre el valor si no se especifica, es aplicable a elementos posicionados, y la soportan todos los navegadores.

```
  #contenido {
    position:absolute;
    z-index: 4;
    }
```

2.6.7. Visibility

Hace que un elemento sea completamente transparente sin eliminarlo del flujo del documento.

Los valores que podemos especificar son:

- **Visible**: el elemento es visible.
- **Hidden**: el elemento es invisible (totalmente transparente), pero sigue afectando a la composición.
- **Collapse**: si se utiliza en elementos que no sean filas o columnas de una tabla, "collapse" tiene el mismo significado que "hidden".

Por defecto, es "visible". Es una propiedad que hereda del padre el valor si no se especifica, es aplicable a todos los elementos y la soportan todos los navegadores.

En el siguiente ejemplo el contenido estaría escondido.

```
  #contenido {
    visibility:hidden;
    }
```

2.7. Propiedades de disposición: fondo

Las propiedades que nos permitirán modificar el fondo de los elementos son las siguientes:

2.7.1. Background-attachment

Determina si una imagen aparece fijada o acompañará el contenido si desplazamos la página haciendo scroll.

Los valores que se pueden especificar pueden ser: "scroll" o "fixed". Por defecto, es "scroll". Es una propiedad que no hereda del padre el valor si no se especifica, es aplicable a elementos, y la soportan todos los navegadores.

```
#contingut {
background-attachment: fixed;
}
```

2.7.2. Background-color

Especifica el color de fondo del contenido del elemento.

Los valores que se pueden especificar pueden ser: un valor rgb, un valor hexadecimal, "transparente", o uno de de los 17 nombres de colores predefinidos. Por defecto, es "transparente". Es una propiedad que no hereda del padre el valor si no se especifica, es aplicable a elementos y la soportan todos los navegadores.

Antes hemos utilizado el background-color para mostrar visualmente el efecto de la anchura del menú. Ahora añadiremos colores de fondo a otras partes de nuestro archivo:

```
body {
padding-left: 5px;
font-family: Georgia, "Times New Roman", Serif;
color: #333333;
background-color: #d6d6d6;
font-size: 11px;
}

h1 {
font-family: Helvetica, Geneva, Arial, Sans-serif;
}

h2 {
color:#3366FF;
}
```

```
#cabecera {
padding-left:5px;
}

#menu {
float: left;
width: 30%;
background-color:#e5e5e5;
padding-top:5px;
padding-bottom:5px;
padding-left: 5px;
}

#contenido {
margin: 30px 10px 0px 32%;
padding-left:5px;
}

#contenido table {
border:1px solid #3366FF;
}

#contenido address {
margin-top:10px;
}

#contenido img {
float:left;
padding-right:10px;
padding-bottom:5px;
}
```

estils.css - Bloc de notas

Archivo Edición Formato Ver Ayuda

```
body {
 padding-left: 5px;
 font-family: Georgia,"Times New Roman", serif;
 color: #333333;
 background-color: #d6d6d6;
 font-size: 11px;
}


h1 {
 font-family: Helvetica, Geneva, Arial, sans-serif;
}
```

Si visualizamos el código anterior en un navegador veremos que el efecto es bastante evidente:



2.7.3. Background-image

Con esta propiedad podemos especificar una imagen de fondo para un elemento.

Los valores que se pueden especificar pueden ser: una dirección URL o "none". Por defecto es "none". Es una propiedad que no hereda del padre el valor si no se especifica, es aplicable a elementos y la soportan todos los navegadores.

```
#contenido {
  background-image:url(/imgs/mural.jpg);
  }
```

2.7.4. Background-position

Especifica la posición inicial de la imagen de fondo del elemento.

Como valores, podemos especificar una posición horizontal seguida de una posición vertical. Cada una de las dos posiciones las podemos especificar indicando una medida, un porcentaje o una de las 5 palabras clave de posicionamiento. Por defecto es "0 0".

Es una propiedad que no hereda del padre el valor si no se especifica, es aplicable a elementos y la soportan todos los navegadores.

```
#contenido {
  background-position:left top;
  }
```

2.7.5. Background-repeat

Determina de qué manera la imagen de fondo se repite en el elemento.

Los valores que podemos especificar son: repeat, repeat-x, repeat-y, y no-repeat.

- **repeat**: la imagen se repite horizontal y verticalmente.
- **repeat-x**: la imagen se repite sólo horizontalmente.
- **repeat-y**: la imagen se repite sólo verticalmente.
- **no-repeat**: la imagen no se repite.

Por defecto es "repeat".

Es una propiedad que no hereda del padre el valor si no se especifica, es aplicable a elementos y la soportan todos los navegadores.

```
#contenido {
  background-repeat:repeat-x;
  }
```

2.7.6. Background

Nos permite especificar las 5 propiedades del background al mismo tiempo.

Los valores a especificar son los valores de las propiedades de background-attachment background-color background-image background-position background-repeat. Si se omite algún valor, se aplicará el valor por defecto de la propiedad.

Es una propiedad que no hereda del padre el valor si no se especifica, es aplicable a elementos y la soportan todos los navegadores.

```
#contenido {
  background: fixed url(/imgs/ad.png) left top no repeat;
  }
```

2.8. Elementos

En este capítulo, veremos las propiedades específicas de ciertos elementos. Básicamente, las listas y las tablas.

2.8.1. Listas

Comenzamos con las propiedades específicas que podemos dar a las listas de un documento (ul, ol, dl).

List-style-type

Especifica el tipo de marcador de los ítems de la lista.

Los valores que podemos especificar son:

- **disc**: un disco.
- **circle**: un círculo.
- **square**: un cuadrado.
- **decimal**: números decimales empezando por 1.
- **decimal-leading-zero**: números decimales empezando por 0.
- **lower-roman**: números romanos en minúsculas.
- **upper-roma**: números romanos en mayúsculas.
- **lower-greek**: letras griegas en minúsculas (alfa/α, beta/β, gamma/γ).
- **lower-latin**: letras ASCII en minúsculas.
- **upper-latin**: letras ASCII en mayúsculas.
- **armenian**: numeración armenia tradicional (ayb/ayp, ben/pen, gim/keem...).
- **georgian**: numeración georgiana tradicional
- **lower-alpha**: igual que lower-latin
- **upper-alpha**: igual que upper-latin.
- **none**: no saldrá ningún marca.

El valor por defecto es "disc".

Es una propiedad que hereda del padre el valor si no se especifica, es aplicable a elementos de una lista y la soportan todos los navegadores.

```
li {
  list-style-type:georgian;
  }
```

Cambiaremos, en nuestro ejemplo, el estilo de la visualización de la lista ordenada; para hacerlo, nos referiremos a los elementos ol dentro de la capa menú:

```
body {
padding-left: 5px;
```

```
font-family: Georgia, "Times New Roman", Serif;
color: #333333;
background-color: #d6d6d6;
font-size: 11px;
}

h1 {
font-family: Helvetica, Geneva, Arial, Sans-serif;
}

h2 {
color:#3366FF;
}

#cabecera {
padding-left:5px;
}

#menu {
float: left;
width: 30%;
background-color:#e5e5e5;
padding-top:5px;
padding-bottom:5px;
padding-left: 5px;
}

#menu ol {
list-style-type:circle;
}

#contenido {
margin: 30px 10px 0px 32%;
padding-left:5px;
}

#contenido table {
border:1px solid #3366FF;
}

#contenido address {
margin-top:10px;
}

#contenido img {
float:left;
padding-right:10px;
```

```
    padding-bottom:5px;
    }
```

```
estils.css - Bloc de notas
Archivo  Edición  Formato  Ver  Ayuda
 background-color:#e5e5e5;
 padding-top:5px;
 padding-bottom:5px;
 padding-left: 5px;
}

#menu ol {
 list-style-type:circle;
}

#contingut {
 margin: 30px 10px 0px 32%;
 padding-left:5px;
```

Si lo visualizamos en un navegador, veremos lo siguiente:



Podemos comprobar cómo la lista ordenada del menú que antes aparecía con números ahora nos aparece con círculos.

List-style-position

Especifica la posición del marcador de la lista en relación con la caja principal.

Los valores que se pueden especificar son:

- **outside**: el marcador se sitúa fuera de la caja principal

- **inside**: el marcador queda en el interior de la caja junto con el texto.

El valor por defecto es "outside".

Es una propiedad que hereda del padre el valor si no se especifica, es aplicable a todos los elementos de una lista y la soportan todos los navegadores.

```
li {
  list-style-type:georgian;
  }
```

List-style-image

Especifica la imagen que se utilizará como marcador.

Los valores que se pueden especificar son o bien una dirección URL o bien "none". Por defecto será "none".

Es una propiedad que hereda del padre el valor si no se especifica, es aplicable a todos los elementos de una lista y la soportan todos los navegadores.

```
li {
  list-style-image:url(/imgs/ad.png);
  }
```

List-style

Nos permite especificar las propiedades individuales del "list-style" al mismo tiempo.

Los valores a especificar son los valores de las propiedades de list-style-type list-style-position list-style-image. Si se omite algún valor, se aplicará el valor por defecto de la propiedad.

Es una propiedad que hereda del padre el valor si no se especifica, es aplicable a todos los elementos de una lista, y la soportan todos los navegadores.

```
li {
  list-style:decimal-leading-zero outside url(/imgs/ad.png);
  }
```

2.8.2. Tablas

Las propiedades que podemos modificar específicamente en el caso de las tablas son las siguientes:

Border-collapse

Determina si las celdas de tablas tienen bordes separados o comparten los bordes con las celdas adyacentes, grupos de filas y columnas o con la misma tabla.

Los valores que se pueden especificar son: "collapse", o "separate". Por defecto, será "separate". Es una propiedad que hereda del padre el valor si no se especifica, es aplicable a elementos de tipo table y la soportan todos los navegadores.

```
table{
  border-collapse: separate
  }
```

Border-spacing

Determina el espacio entre las parejas de bordes.

Podemos darle uno o dos valores numéricos. Si los dos valores están presentes, el primero determina la distancia horizontal y el segundo la distancia vertical. El valor por defecto será "0".

Es una propiedad que hereda del padre el valor si no se especifica, es aplicable a todos los elementos de tipo table con bordes separados, y la soportan todos los navegadores.

Daremos a nuestro ejemplo un espacio en los bordes de nuestra tabla. Con el fin de hacerlo añadiremos el siguiente código:

```
body {
padding-left: 5px;
font-family: Georgia, "Times New Roman", Serif;
color: #333333;
background-color: #d6d6d6;
font-size: 11px;
}

h1 {
font-family: Helvetica, Geneva, Arial, Sans-serif;
}

h2 {
color:#3366FF;
}

#cabecera {
padding-left:5px;
}
```

```
#menu {
float: left;
width: 30%;
background-color:#e5e5e5;
padding-top:5px;
padding-bottom:5px;
padding-left: 5px;
}

#menu ol {
list-style-type:circle;
}

#contenido {
margin: 30px 10px 0px 32%;
padding-left:5px;
}

#contenido table {
border:1px solid #3366FF;
border-spacing:10px 10px;
}

#contenido address {
margin-top:10px;
}

#contenido img {
float:left;
padding-right:10px;
padding-bottom:5px;
}
```

estils.css - Bloc de notas

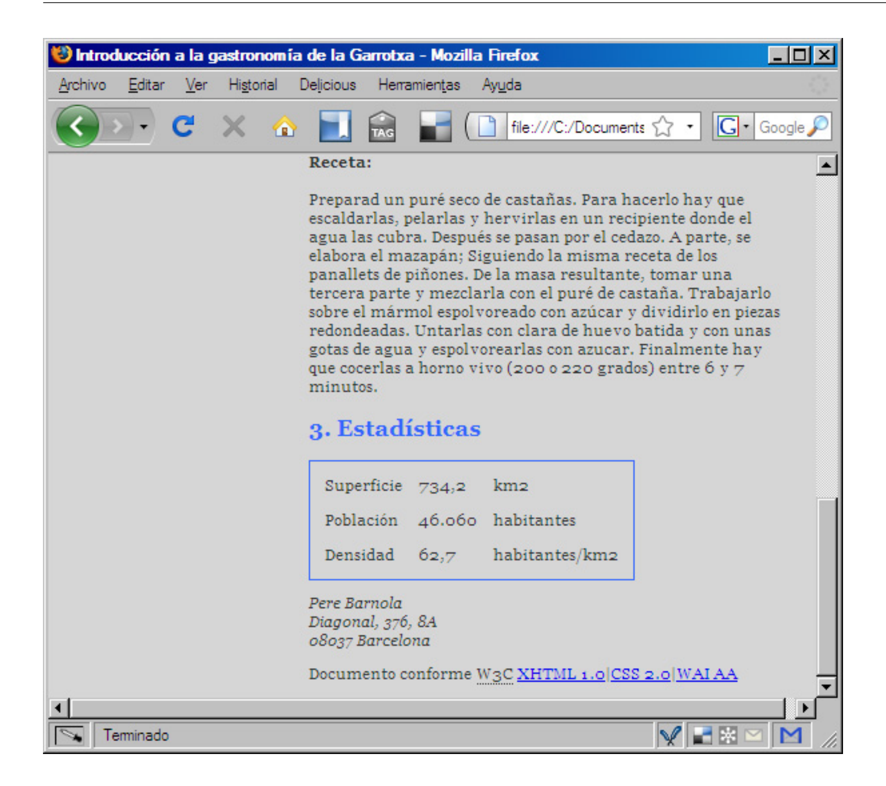
Archivo Edición Formato Ver Ayuda

```
#contingut {
 margin: 30px 10px 0px 32%;
 padding-left:5px;
}

#contingut table {
 border:1px solid #3366FF;
 border-spacing:10px 10px;
}

#contingut address {
 margin-top:10px;
```

Y lo visualizaremos de la siguiente manera:



Vemos cómo ahora hay más espacio entre el contenido y las celdas de nuestra tabla.

Empty-cells

Oculta o muestra los bordes de las celdas vacías.

Los valores pueden ser: "show" o "hide". Por defecto, el valor será "show".

Es una propiedad que hereda del padre el valor si no se especifica, es aplicable a todos los elementos de tipo "table" con bordes separados y la soportan todos los navegadores excepto Internet Explorer.

Table-layout

Determina qué algoritmo hay que utilizar para el navegador a la hora de visualizar las celdas, filas y columnas de una tabla.

Los valores pueden ser: "auto" o "fixed".

- **fixed**: permite que el navegador renderice más rápidamente el contenido de la tabla que de la forma automática. La presentación horizontal sólo depende de la anchura de la tabla y las columnas, no del contenido de la tabla. Utilizando este método se puede visualizar la tabla una vez la primera fila se ha descargado correctamente.

- **auto**: la anchura de la columna se determina por la anchura mayor del contenido indivisible dentro de la celda. El navegador tiene que leer toda la tabla antes de mostrarla.

Por defecto, el valor será "auto".

Es una propiedad que hereda del padre el valor si no se especifica, es aplicable a todos los elementos de tipo "table" que no tengan la propiedad "width" especificada como "auto", y la soportan todos los navegadores excepto Internet Explorer.

```
table{
  table-layout: fixed
  }
```

Caption-side

Determina la posición del título de la tabla, si éste va encima o debajo de la tabla.

Los valores pueden ser: "top" o "bottom". Por defecto, el valor será "top".

Es una propiedad que hereda del padre el valor si no se especifica lo contrario, es aplicable a todos los elementos de tipo tabla, y la soportan todos los navegadores excepto Internet Explorer.

En el caso de las tablas, hay propiedades que ya hemos visto anteriormente que tienen un efecto especial cuando se aplican a tablas o celdas. Son el "width" y el "vertical-align".

Width

Determina una anchura mínima de una tabla o de una celda.

Los valores pueden ser un número, un porcentaje, o "auto". Por defecto será "auto".

Es una propiedad que no hereda del padre el valor si no se especifica, es aplicable a elementos de tipo table, table-column y table-cell y la soportan todos los navegadores.

Vertical-align

Determina la alineación vertical del texto dentro de la fila o celda.

Los valores pueden ser: "baseline", "top", "bottom", o "middle". Por defecto será "baseline".

Es una propiedad que no hereda del padre el valor si no se especifica, es aplicable a elementos de tipo "table-cell", y la soportan todos los navegadores.

2.9. Otros

2.9.1. Herencia

Una de las características más importantes del CSS es la que se conoce como herencia u orden en cascada. Lo que significa trabajar en cascada se puede entender fácilmente a partir del ejemplo siguiente.

Imaginad que queréis crear un documento que tenga toda la tipografía de color gris, de una medida de 12 píxeles, sobre un fondo claro. Para conseguirlo, tendréis que definir la regla siguiente:

```
*{
  color:#33ff00;
  background-color:#eeeeee;
  font-size:12px;
  }
```

Esta regla de selector universal (el asterisco) afectará a todos los elementos del documento. Sin embargo, poned por caso que no queréis que la cabecera del primer nivel tenga el mismo color que el resto de elementos, sino que lo que queréis es que sea de color rojo. Por eso habrá que añadir una segunda regla:

```
h1 {
  color: red;
  }
```

El navegador sobrescribirá el valor del color selector h1 sobre el selector universal, pero el selector h1 heredará el resto de propiedades de la regla más general, y mostrará, por lo tanto, la misma medida y el mismo color de fondo que el resto de elementos.

Se puede decir que las diferentes reglas compiten entre ellas según un orden de jerarquía o prioridad, de manera que las propiedades de una regla de un orden superior prevalecerán siempre sobre las de un orden inferior.

Este orden en cascada separa las reglas en seis grupos diferentes según el tipo de selector utilizado. El listado de los seis grupos está ordenado de mayor a menor prioridad. En términos generales, tendrán siempre preferencia los tipos de selectores de más precisión frente a los selectores más generales.

Los seis grupos en que quedan agrupadas las reglas, según el tipo de selector utilizado, ordenados de mayor a menor prioridad, son los siguientes:

Propiedades que contengan la expresión "!important" detrás de su valor:

```
h1{
  color:red !important;
  }
```

Estilos declarados como valor del atributo "style" del elemento XHTML:

```
<h1 style="color red;">Lorem Ipsum</p>
```

Reglas definidas por uno o más selectores del tipo ID:

```
#vermell {
  color: red;
  }
```

Reglas definidas por uno o más clases, atributos o pseudoselectores:

```
  . rojo {
  color:red;
  }

  p:first-letter {
  color: red;
  }
```

Reglas que contienen uno o más selectores XHTML:

```
  h1 {
  color: red;
  }
```

Reglas que contienen el selector universal:

```
  * {
  color: red;
  }
```

2.10. Caso práctico: "dando estilos a nuestra primera web"

A partir de los conocimientos que hemos adquirido, daremos estilo al documento de ejemplo que construimos en el segundo capítulo mediante las hojas de estilo.

La idea es que partáis del código XHTML y CSS que tenéis a continuación, y modifiquéis las reglas para ir observando cómo los diferentes valores que hemos visto en cada una de las propiedades afectan a la visualización del documento.

Archivo XHTML llamado "index_estilos.html"

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="es" lang="es">
<head>
<meta http-equiv="Content-Type" content="text/html; charset=UTF-8" />
<meta name="author" content="Pere Barnola"/>
<meta name="keywords" content="garrotxa, gastronomía"/>
<title>Introducción a la gastronomía de la Garrotxa</title>
<link href="estilos.css" rel="stylesheet" type="text/css" />
</head>
<body>
<div id="cabecera">
<h1>La gastronomía en la Garrotxa</h1>
</div>
<div id="menu">
<h1>Menú</h1>
<ol>
<li><a href="#introduccion" title="Descripción del taller">Introducción</a></li>
<li><a href="#platos_tipicos" title="Conocimientos a adquirir">Platos típicos</a></li>
<li><a href="#estadisticas" title="Guías que se utilizan">Estadísticas</a></li>
</ol>
</div>
<div id="contenido">
<h2>1. Introducción<a name="introduccion"></a></h2>
<img src="foto_garrotxa.gif"
alt="imagen del cráter de Santa Margalida" width="376" height="226" />
<p>La Garrotxa una comarca de transicitre la montaña y la planallanura,
que se estentre el Pirineo y la Cordillera Transversal. Aun la
complejidad de la comarca se distinguen dos sectores bien
diferenciados y delimitados: la Alta Garrotxa y la Baja Garrotxa.
La primera, ferega y de relieve difícil, con poca vegetación llena de
desfiladeros, simas y cestos, da nombre a la comarca -Garrotxa, 'tierra
áspera y rota'-, y un paradper a los excursionistas; la Baja
Garrotxa, de sierras suaves y depresiones volcques, mplanera, con
prados abundantes y corrientes de agua</p>
<h2>2. Platos típicos<a name="platos_tipicos"></a></h2>
<h3>PANECILLOS DE CASTAÑA</h3>
<h4>Ingredientes:</h4>
<ul>
```

```
<li> 500 gr de almendras ralladas</li>
<li> 450 gr de azúcar</li>
<li> 200 gr de patatas</li>
<li> 25 gr de azúcar lustre</li>
<li> 2 huevos</li>
<li> 250 gr de castañas</li>
</ul>
<h4>Receta:</h4>
<p>Preparad un puré seco de castañas. Para hacerlo, hay que
escaldarlas, pelarlas y hervirlas en un recipiente que las cubra
con agua. Después, las pasáis por el cedazo. Aparte, elaborad el
mazapán; siguiendo la misma receta de los panecillos de piñones.
De la masa resultante, tomad una 1/3 parte y mezcladla con
el puré de castaña. Trabajadlo sobre el mármol empolvado con
azúcar y divididlo en piezas redondeadas. Untadlas con clara
de huevo batida con unas gotas de agua y empolvadlas con azúcar.
Finalmente, hay que cocerlas en el horno (200 o 220 grados) entre
6 o 7 minutos.</p>
<h2>3. Estadísticas<a name="estadisticas"></a></h2>
<table>
<tr>
<td>Superficie</td>
<td>734,2</td>
<td>km2</td>
</tr>
<tr>
<td>Población</td>
<td>46.060</td>
<td>habitantes</td>
</tr>
<tr>
<td>Densidad</td>
<td>62,7</td>
<td>habitantes/km2</td>
</tr>
</table>
<address>
Pere Barnola<br />
Diagonal, 376, 8A<br />
08037 Barcelona
</address>
<p>Documento conforme <acronym title="World Wide Web Consortium"
xml:lang="en">W3C</acronym> <a href="http://validator.w3.org/"
title="Código revisado con el validador del W3C">XHTML 1.0</a>|
<a href="http://jigsaw.w3.org/css-validator/" title="Hojas de
 estilo revisadas con el validador del W3C">CSS 2.0</a>|
<a href="http://www.tawdis.net/taw3/online"
```

```
 title="Nivel de adecuación doble A, conforme a las
 directrices de la Web Accessibility Initiative">WAI AA</a></p>
</div>
</body>
</html>
```

Código del archivo estilos.css resultante

Validación final

Finalmente, una vez hechas las modificaciones, hará falta que os aseguréis de que el documento CSS sea un documento CSS válido mediante el Servicio de Validación de CSS del W3C.

```
body {
padding-left: 5px;
font-family: Georgia, "Times New Roman", Serif;
color: #333333;
background-color: #eeeeee;
}

h1 {
font-family: Helvetica, Geneva, Arial, Sans-serif;
}

h2 {
color:#3366FF;
border-bottom:1px solid #3366FF;
}
a {
color:#3366FF;
}
a:hover {
color:#33CCFF;
}

#cabecera {
border: dotted 1px #33CCFF;
padding-left:5px;
}

#menu {
float: left;
width: 30%;
background: #eeeeee;
padding-top:5px;
padding-bottom:5px;
padding-left: 5px;
background-color:#e5e5e5;
}
#menu ol {
list-style-type:circle;
}
```

```
#contenido {
margin: 30px 10px 0px 32%;
padding-left:5px;
}

#contenido table {
border:1px solid #3366FF;
border-spacing:10px 10px;
}

#contenido address {
margin-top:10px;
color: #999999;
font-size:11px;
}

#contenido img {
float:left;
padding-right:10px;
padding-bottom:5px;
}

#contenido ul {
list-style-type: square;
}
```